2025 IEEE 33rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM) | 979-8-3315-0281-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/FCCM62733.2025.00059

DRSA: Accelerating Macro Placement on Commercial FPGAs

Menzo Bouaissi*, Paolo Ienne*, Lana Josipović‡, and Andrea Guerrieri†*
*EPFL, School of Computer and Communication Sciences, Lausanne, Switzerland
‡ETH Zurich, Department of Information Technology and Electrical Engineering, Zurich, Switzerland
†HES-SO Valais-Wallis, School of Engineering, Sion, Switzerland

I. Introduction

FPGAs are highly versatile devices, but their backend compilation process is significantly time-consuming. DynaRapid has demonstrated the potential to drastically reduce compilation times to mere seconds by leveraging macro-component-based design hierarchies [1]. However, DynaRapid faces challenges in macro-component placement, often resulting in frequency degradation. In this work, we propose DRSA, a fast placer based on simulated annealing targeting DynaRapid's macros, capable of overcoming the frequency degradation of the previous greedy placement strategy [1].

II. Accelerating Macro Placement on FPGA

Simulated Annealing (SA) is used in DRSA to optimize the placement of macro components by minimizing a cost function. Inspired by ExHiPR [2], the cost function is:

$$\alpha WL_{norm} + \beta D_{norm} + \gamma AS_{norm} + \sigma OV_{norm}. \quad (1)$$

The weights α, β, γ are positive values that sum up to 1 ($\alpha + \beta + \gamma = 1$), while σ is a parameter greater than or equal to 1. An illegal placement is detectable if the cost function is greater than 1, indicating that overlap across macros occurred [2]. This baseline, referred to as BSA, converges to the critical path of Vivado, yet at significant runtime expense ($2\times$).

A. Speeding Up the Convergence

To speed up convergence, one effective approach is to narrow the search space of potential placement locations. To reduce the search space during random movement, we employ a Gaussian multivariate distribution to dynamically increase the likelihood of positioning the design within a specific region. We also raise the cooling rate, though this may come at a solution quality cost.

B. Improving The Critical Path

To address critical path degradation introduced by the increased cooling rate, we refine the terms of the cost function. To this end, we introduced a new term to Equation 1: the area of individual shapes (AR). We explored cost functions containing various combinations of terms, including density, aspect ratio, and area. We extend our analysis beyond immediate connections to further improve accuracy in estimating wire length. To capture the impact of placement on wire length, we split the weight α into two factors: first-degree, for direct connections, and second-degree, for next neighbors. Finally, we incorporated a macro-swapping mechanism into the placement strategy. This mechanism resolves macro overlaps by swapping components before the placement is invalidated.

III. Evaluation

We compare our placement strategy with Vivado 2024.1, using `-directive Quick` targeting a Virtex UltraScale+™ xcvu13p. DRSA achieves, on average, a $9.8\times$ faster placement compared to BSA and a $4.9\times$ speedup over the commercial placer. We then compared the placement density of DRSA using Vivado's minimum dimensions (Figure 1), observing that DRSA achieved a comparable density of 65%.

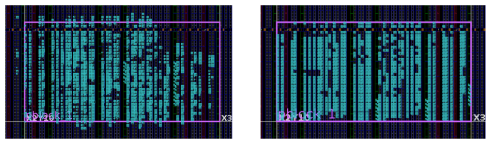

Fig. 1: Visualization of placement density for the *video_filter* benchmark. (a) Vivado minimum bounding box. (b) DRSA placement constrained to Vivado minimum bounding box.

IV. Conclusion

In this work, we present DRSA, a placer based on simulated annealing that overcomes the critical path degradation of the greedy placement strategy of DynaRapid [1], while still outperforming Vivado in terms of runtime. As shown in Figure 2, DRSA enables $3.91\times$ and $7.78\times$ faster runtime than Vivado and BSA, respectively, but also lowers the critical path of 2% on average after routing.

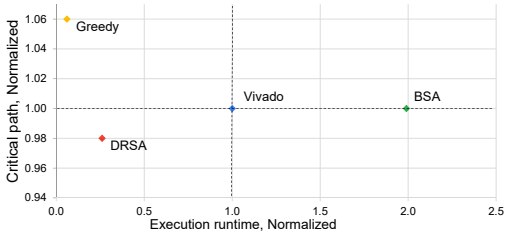

Fig. 2: Comparison of DynaRapid's runtime and critical path, using DRSA, BSA, and greedy placer, normalized on Vivado.

References

[1] A. Guerrieri, S. Guha, C. Lavin, E. Hung, L. Josipović, and P. Ienne, "DynaRapid: Fast-tracking from C to routed circuits," in *34th International Conference on Field-Programmable Logic and Applications*, 2024, pp. 24–32.

[2] Y. Xiao, D. Park, Z. J. Niu, A. Hota, and A. DeHon, "ExHiPR: Extended high-level partial reconfiguration for fast incremental FPGA compilation," *ACM Trans. Reconfigurable Technol. Syst.*, Sep. 2023.

2576-2621/25/$31.00 ©2025 IEEE
DOI 10.1109/FCCM62733.2025.00059